

目次

訳者まえがき	v
まえがき	vii
1 章 はじめに.....	1
1.1 この本で説明する内容.....	1
1.2 なぜ Python はデータ分析者におすすめなのか	2
1.2.1 「糊 (グルー)」としての Python.....	2
1.2.2 「2つの言語を利用する」ことの問題を解決する	2
1.2.3 Python を使わない場合.....	3
1.3 本書で扱う重要な Python ライブラリ	3
1.3.1 NumPy	3
1.3.2 pandas	4
1.3.3 matplotlib	5
1.3.4 IPython.....	5
1.3.5 SciPy.....	5
1.4 インストールとセットアップ	6
1.4.1 Windows	7
1.4.2 Apple OSX.....	9
1.4.3 GNU/Linux.....	10
1.4.4 Python 2 と Python 3 の相違点	11
1.4.5 統合開発環境 (IDE)	11
1.5 コミュニティとカンファレンス (会議)	11
1.6 この本の読み方の案内.....	12
1.6.1 コード例	12
1.6.2 例として用いるデータ.....	12
1.6.3 インポートの決まりごと	13
1.6.4 専門用語 (ジャーゴン、jargon)	13

1.7	謝辞	13
2 章	Python によるデータ分析事例	15
2.1	データ分析の例：短縮 URL 1.usa.gov への変換データ	15
2.1.1	Python 標準機能を使用したタイムゾーン情報の集計	17
2.1.2	pandas を使用したタイムゾーン情報の集計	20
2.2	データ分析の例：MovieLens 1M（映画評価データ）	27
2.2.1	評価の分かれた映画の抽出	32
2.3	データ分析の例：アメリカの赤ちゃんに名付けられた名前リスト（1880-2010）	34
2.3.1	名付けの傾向分析	39
2.4	この章のまとめと次のステップ	50
3 章	IPython：対話的な開発環境	51
3.1	IPython の基本	52
3.1.1	タブ補完	53
3.1.2	イントロスペクション	54
3.1.3	%run コマンド	56
3.1.4	クリップボード経由の実行	57
3.1.5	キーボードショートカット	59
3.1.6	例外とトレースバック	60
3.1.7	マジックコマンド	60
3.1.8	Qt ベースの高機能 GUI コンソール	62
3.1.9	Matplotlib との連携、Pylab モード	63
3.2	コマンド履歴の利用	64
3.2.1	コマンド履歴の検索とその再利用	64
3.2.2	入出力変数	65
3.2.3	入出力のロギング	66
3.3	オペレーティングシステムとの連携	67
3.3.1	シェルコマンドとエイリアス（別名定義）	67
3.3.2	ディレクトリブックマーク機能	69
3.4	ソフトウェア開発ツール	69
3.4.1	対話的デバッガ	70
3.4.2	処理時間の計測：%time と %timeit	75
3.4.3	プロファイリングの基本：%prun と %run -p	76
3.4.4	行ごとのプロファイリング	78
3.5	IPython HTML ノートブック	82
3.6	IPython での生産的コード開発に向けたヒント	84

3.6.1	依存関係を考慮したモジュールの再読み込み	84
3.6.2	コード設計のヒント	85
3.7	高度な IPython 機能	86
3.7.1	自前のクラスの IPython への親和性を高める技法	86
3.7.2	IPython プロファイルと構成機能	87
3.8	謝辞	88
4 章	NumPy の基本：配列とベクトル演算	89
4.1	NumPy ndarray：多次元配列オブジェクト	90
4.1.1	ndarray の生成	91
4.1.2	ndarray 要素のデータ型	93
4.1.3	ndarray とスカラーの計算	96
4.1.4	インデックス参照とスライシングの基礎	97
4.1.5	ブールインデックス参照	101
4.1.6	ファンシーインデックス参照	104
4.1.7	転置行列、行と列の入れ替え	106
4.2	ユニバーサル関数：すべての配列要素への関数適用	107
4.3	ndarray を用いたデータ処理	110
4.3.1	条件制御の ndarray での表現	111
4.3.2	数学関数、統計関数	114
4.3.3	真偽値の配列関数	115
4.3.4	ソート	116
4.3.5	集合関数：unique など	117
4.4	ndarray のファイル入出力	118
4.4.1	ndarray の保存：バイナリ形式	118
4.4.2	ndarray の保存：テキスト形式	119
4.5	行列計算	120
4.6	乱数生成	122
4.7	例：ランダムウォーク	123
4.7.1	多重ランダムウォーク	124
5 章	pandas 入門	127
5.1	pandas のデータ構造	128
5.1.1	シリーズ (Series)	128
5.1.2	データフレーム (DataFrame)	132
5.1.3	インデックスオブジェクト	138
5.2	pandas の重要な機能	140
5.2.1	再インデックス付け	140

5.2.2	軸から要素を削除する	143	7.1.3	軸に沿った連結	214
5.2.3	インデックス参照、選択、フィルタリング	144	7.1.4	重複のあるデータの結合	218
5.2.4	算術とデータの整形	147	7.2	再形成とピボット	220
5.2.5	関数の適用とマッピング	152	7.2.1	階層的にインデックス付けされているデータの再形成	220
5.2.6	ソートとランク	153	7.2.2	「long」フォーマットから「wide」フォーマットへのピボット	222
5.2.7	重複した値を持つ軸のインデックス	156	7.3	データの変換	224
5.3	要約統計量の集計と計算	157	7.3.1	重複の除去	225
5.3.1	相関と共分散	160	7.3.2	関数やマッピングを用いたデータの変換	226
5.3.2	一意な値、頻度の確認、所属の確認	162	7.3.3	値の置き換え	228
5.4	欠損値の取り扱い	164	7.3.4	軸のインデックスの名前を変更する	229
5.4.1	欠損値を除外する	165	7.3.5	離散化とビンニング	230
5.4.2	欠損値を穴埋めする	167	7.3.6	外れ値の検出と除去	233
5.5	階層型インデックス	169	7.3.7	並べ替えとランダムサンプリング	234
5.5.1	階層の順序変更とソート	172	7.3.8	指標やダミー変数の計算	235
5.5.2	階層ごとの要約統計量	173	7.4	文字列操作	238
5.5.3	データフレームの列をインデックスに使う	173	7.4.1	String オブジェクトのメソッド	238
5.6	pandas のその他のトピック	175	7.4.2	正規表現	240
5.6.1	整数を使ったインデックス参照	175	7.4.3	pandas における、ベクトル化された文字列関数	244
5.6.2	パネル	176	7.5	例：USDA の食品データベース	246
6 章	データの読み込み、書き出しとファイル形式	179	8 章	プロットと可視化	253
6.1	テキスト形式のデータの読み書き	179	8.1	matplotlib API の概要	253
6.1.1	テキストファイルを少しずつ読み込む	185	8.1.1	図とサブプロット	254
6.1.2	テキスト形式でのデータの書き出し	187	8.1.2	色、マーカー、線種	258
6.1.3	区切り文字で区切られた形式を手で操作する	189	8.1.3	目盛り、ラベル、凡例	260
6.1.4	JSON データ	191	8.1.4	注釈やサブプロットへの描画	263
6.1.5	XML と HTML : Web スクレイピング	192	8.1.5	プロットのファイルへの保存	265
6.2	バイナリデータ形式	197	8.1.6	matplotlib の設定	266
6.2.1	HDF5 形式の使用	198	8.2	pandas のプロット関数	267
6.2.2	Microsoft Excel ファイルの読み込み	199	8.2.1	折れ線グラフ	267
6.3	HTML や Web API を用いた読み書き	199	8.2.2	棒グラフ	269
6.4	データベースの読み書き	201	8.2.3	ヒストグラムと密度プロット	273
6.4.1	MongoDB へのデータの保存や読み取り	203	8.2.4	散布図	275
7 章	データの管理：データのクリーニング、変換、マージ、再形成	205	8.3	地図のプロット：ハイチ地震災害のデータ	276
7.1	データセットのマージ	205	8.4	Python 可視化ツールのエコシステム	283
7.1.1	データベース風のデータフレームのマージ	206	8.4.1	Chaco	283
7.1.2	インデックスによるマージ	210	8.4.2	MayaVi	284
			8.4.3	その他のパッケージ	284

8.4.4	可視化ツールの未来は？	284	10.4	タイムゾーンを扱う	345
9 章	データの集約とグループ演算	285	10.4.1	ローカライゼーションと変換	346
9.1	GroupBy の仕組み	286	10.4.2	タイムゾーンを意識したタイムスタンプオブジェクト	348
9.1.1	グループをまたいだ繰り返し	289	10.4.3	別のタイムゾーンとの演算	349
9.1.2	列や列の集合の選択	291	10.5	期間を使った算術演算	349
9.1.3	ディクショナリ形式やシリーズのグルーピング	292	10.5.1	期間頻度の変換	351
9.1.4	関数を用いたグルーピング	293	10.5.2	四半期の頻度	352
9.1.5	インデックスレベルによるグルーピング	294	10.5.3	タイムスタンプから期間への変換（とその逆）	354
9.2	データの集約	295	10.5.4	配列から PeriodIndex を作成する	355
9.2.1	列に複数の関数を適用する	297	10.6	再サンプリングと頻度変換	356
9.2.2	集約されたデータを「インデックス付けされていない」形式に戻す	300	10.6.1	ダウンサンプリング	357
9.3	グループ指向の演算と変形	300	10.6.2	アップサンプリングと穴埋め	360
9.3.1	apply メソッド： 一般的な分離 - 適用 - 結合（split-apply-combine）の方法	302	10.6.3	期間で再サンプリングする	362
9.3.2	分位点とビン分析	305	10.7	時系列のプロット	363
9.3.3	例：グループごとに指定する値で欠損値を埋める	306	10.8	移動する窓関数	366
9.3.4	例：ランダムサンプリングと順列	308	10.8.1	指数加重関数	368
9.3.5	例：グループの加重平均と相関	310	10.8.2	2 値の場合での移動する窓関数	369
9.3.6	例：グループ指向の線形回帰	312	10.8.3	ユーザ定義の移動する窓関数	370
9.4	ピボットテーブルとクロス集計	313	10.9	パフォーマンスとメモリ利用の注意点	371
9.4.1	crosstab メソッド：クロス集計	315	11 章	金融と経済データへの応用	373
9.5	例：2012 年の連邦選挙委員会のデータベース	316	11.1	データ変更に関するトピック	373
9.5.1	職業と勤務先による寄付金の統計	319	11.1.1	時系列とクロスセクションの整形	373
9.5.2	寄付金額のビン分割	323	11.1.2	異なる頻度の時系列に対する演算	376
9.5.3	州別の寄付金の統計	325	11.1.3	時刻とその時点でのデータ取得	379
10 章	時系列データ	329	11.1.4	データソースをつなぎ合わせる	381
10.1	日付、時間のデータ型とツール	330	11.1.5	リターンインデックスと累積リターン	384
10.1.1	文字列と datetime の変換	331	11.2	グループの変換と分析	386
10.2	時系列の基本	334	11.2.1	因子への露出度のグルーピング	388
10.2.1	インデックス参照、データの選択、サブセットの抽出	335	11.2.2	分位分析	389
10.2.2	重複したインデックスを持つ時系列	337	11.3	その他の応用例	391
10.3	日付範囲、頻度、シフト	338	11.3.1	シグナル境界分析	391
10.3.1	日付範囲の生成	339	11.3.2	先物取引契約のロールオーバー	394
10.3.2	頻度と日付オフセット	340	11.3.3	移動相関と線形回帰	397
10.3.3	データの前方向と後方へのシフト	343	12 章	NumPy：応用編	399
			12.1	ndarray オブジェクトの内部構造	399
			12.1.1	NumPy dtype の階層構造	400

12.2	配列操作：応用編	401
12.2.1	配列の再形成	401
12.2.2	C と Fortran の順序の違い	403
12.2.3	配列の結合と分割	404
12.2.4	要素の繰り返し：tile と repeat	407
12.2.5	ファンシーインデックス参照の別法：take と put.....	408
12.3	ブロードキャスト	410
12.3.1	他の軸へのブロードキャスト	413
12.3.2	ブロードキャストでの配列への値の設定.....	415
12.4	ufunc の使い方：応用編.....	416
12.4.1	ufunc インスタンスメソッド	416
12.4.2	独自定義の ufunc.....	419
12.5	構造化配列とレコード配列	420
12.5.1	ネストした dtype と多次元フィールド	421
12.5.2	構造化配列を使うべき理由	422
12.5.3	構造化配列の操作：numpy.lib.recfunctions	423
12.6	ソートについてさらに詳しく	423
12.6.1	間接ソート：argsort と lexsort	424
12.6.2	使用可能な他のソートアルゴリズム	426
12.6.3	numpy.searchsorted：ソートされた配列内で要素を探す	427
12.7	NumPy の行列クラス	428
12.8	配列の入出力：応用編.....	430
12.8.1	メモリマップファイル	430
12.8.2	HDF5 やその他の配列保存方法.....	432
12.9	パフォーマンスに関する小技	432
12.9.1	連続したメモリの重要性	432
12.9.2	高速化のためのさらなる選択肢：Cython、f2py、C.....	434

索引.....	437
---------	-----